

## CIRCUITOS LÓGICOS

### Relatório VII

Discente: Rebeca de Macêdo Ferreira.

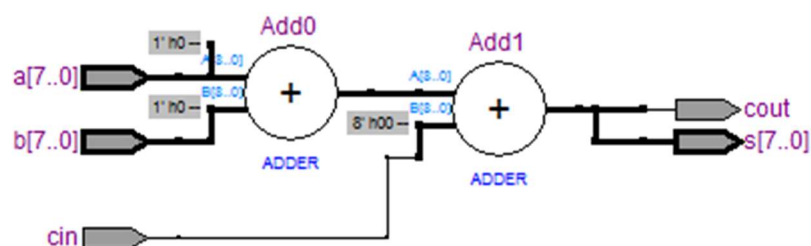
Matrícula: 2016000524

Nesse projeto fizemos um somador de 16 bits

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_UNSIGNED.ALL;
4
5  entity adder is
6      generic (N: integer :=8);
7      port (a, b: in STD_LOGIC_VECTOR(N-1 downto 0);
8          cin: in STD_LOGIC; s: out STD_LOGIC_VECTOR(N-1 downto 0);
9          cout: out STD_LOGIC);
10
11  end;
12
13  architecture synth of adder is
14      signal result: STD_LOGIC_VECTOR(N downto 0);
15
16  begin
17      result <= ('0' & a) + ('0' & b) + cin;
18      s <= result (N-1 downto 0);
19      cout <= result (N);
20  end;
```

Este VHDL foi copiado no livro Digital Design and Computer Architecture, e define como entrada 'a', 'b' e 'cin' e como saída 's' 'cout'.

Após compilado, selecionamos RTL viewer:



Para fazer as simulações corretas, fez-se um test vector

```

1 0000000000000000 1111111111111111_0_0_1111111111111111
2 1111111111111111_0000000000000000_0_0_1111111111111111
3 1111111111111111_0000000000000001_0_1_0000000000000000
4 1111111111111111_1111111111111111_0_1_1111111111111111
5 0000000000000000_1111111111111111_1_1_0000000000000000
6 1111111111111111_0000000000000000_1_1_0000000000000000
7 1111111111111111_0000000000000001_1_1_0000000000000001
8 1111111111111111_1111111111111111_1_1_1111111111111111

```

Os primeiros 16 bits (esquerda pra direita) são referentes à entrada 'a', os próximos 16 bits correspondem à entrada 'b', o bit seguinte é o 'cin', depois o 'cout', depois a saída 's'.

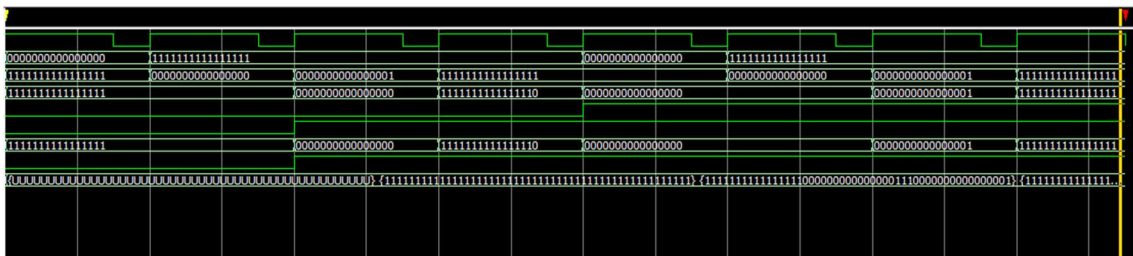
Alteramos o testbench do projeto de um somador completo para o caso do projeto atual...

```

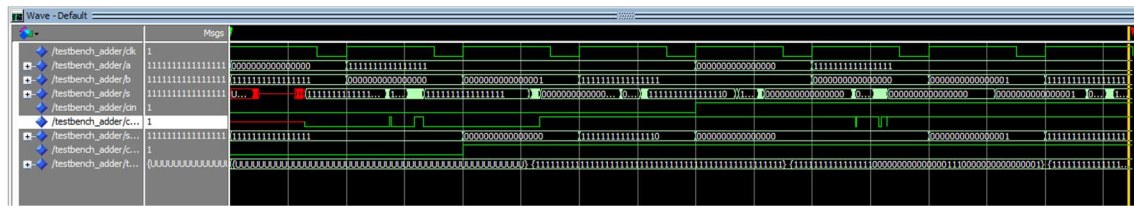
11 architecture sim of testbench_adder is
12     component adder
13     port (a, b: in STD_LOGIC_VECTOR(15 downto 0);
14           cin: in STD_LOGIC;
15           s: out STD_LOGIC_VECTOR(15 downto 0);
16           cout: out STD_LOGIC);
17     end component;
18
19     signal clk: STD_LOGIC;
20     signal a, b, s: STD_LOGIC_VECTOR(15 downto 0);
21     signal cin, cout: STD_LOGIC;
22     signal sexpected: STD_LOGIC_VECTOR(15 downto 0);
23     signal coutexpected: STD_LOGIC;
24     constant MEMSIZE: integer := 8;
25     type tarray is array (MEMSIZE downto 0) of STD_LOGIC_VECTOR (49 downto 0);
26     signal testvectors: tarray;
27     shared variable vectornum, errors: integer;
28     begin
29         -- instantiate device under test
30         dut: adder port map (a, b, cin, s, cout);
31         -- generate clock
32     process begin
33         clk <= '1'; wait for 15 ns;
34         clk <= '0'; wait for 5 ns;
35     end process;

```

O próximo passo foi simular o RTL level



Depois simulamos o Gate level



## DIRETÓRIO

Código VHDL: adder/adder.vhd

Código Testbench: adder/testbench/testbench\_adder.vhd

Arquivo test vector: adder/ simulation / ModelSim / adder.tv